Tecnológico de Monterrey

Análisis de Grandes Volúmenes de Datos

Avance de Proyecto 2:

Sistema de Recomendación

Equipo 39

Luis Gerardo Barbosa Mendoza | A01731203

Miguel Ángel Marines Olvera | A01705317

Genaro Rodríguez Vázquez | A01150931

Luis Ángel Seda Marcos | A01795301

26 / 05 / 2024

1. Exploración del Conjunto de Datos

- Descripción del Conjunto de Datos

  El conjunto de datos contiene información sobre películas y es útil para desarrollar un programa de recomendación de películas. La información fue obtenida de TMDB y GroupLens y contiene información de películas lanzadas hasta julio del 2017.

  El dataset consta de tres archivos en formato CSV. El primer archivo es "credits.csv" e incluye los créditos de las películas y la información de los elencos y equipos de producción. El segundo archivo es “keywords.csv" y proporciona palabras clave relacionadas con las tramas de las películas. El tercer archivo es “movies.csv" y ofrece detalles adicionales sobre cada película.

  El dataset se obtuvo de Kaggle, de la siguiente liga:
  https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?select=keywords.csv

  Nota: El proceso descrito a continuación se puede ver a detalle en la siguiente liga de Google Colab:
  https://colab.research.google.com/drive/1bcVz_nOkUFUtlEYD-u2Ihtq7wgQMGZaa?usp=sharing

  A continuación se describen los archivos del dataset y lo que contienen.

  Credits:

  Contiene los créditos de la película y tiene la información del elenco (cast) y equipo de producción (crew).

  Contienen 3 columnas o variables:

Nombre	**Tipo de Dato**
Cast (Elenco de la película)	Object
Crew (Equipo de producción de la película)	Object
Movie ID (ID de la película)	Int64

  Keywords:

  Contiene palabras clave (keywords) de la trama de la película.

  Contienen 2 columnas o variables:

Nombre	Tipo de Dato
Keywords (Palabras clave de la trama de la película)	Object
Movie ID (ID de la película)	Int64

Movies:

Contiene información adicional y detallada de la película.

Contienen 24 columnas o variables:

Nombre	Tipo de Dato
Adult (Película para adultos)	Object
Belongs to Collection (Pertenece a una saga la película)	Object
Budget (Presupuesto de la película)	Object
Genres (Géneros de la película)	Object
Homepage (Página web de la película)	Object
Movie ID (ID de la película)	Object
IMDB ID (Identificador único de la película)	Object
Original Language (Idioma original de la película)	Object
Original Title (Título original de la película)	Object
Overview (Sinopsis de la película)	Object
Popularity (Popularidad de la película)	Object

Poster Path (Link a poster de la película)	Object
Production Companies (Compañías productoras de la película)	Object
Production Countries (Países donde se produjo la película)	Object
Release Date (Fecha de estreno de la película)	Object
Revenue (Ganancia económica de la película)	Float64
Runtime (Duración de la película)	Float64
Spoken Languages (Idiomas en la que se encuentra disponible la película)	Object
Status (Estatus de la película cómo estrenada, en producción, etc.)	Object
Tagline (Slogan de la película)	Object
Title (Título de la película)	Object
Video (Archivo de Video)	Object
Vote Average (Calificación promedio que recibe la película)	Float64
Vote Count (Total de calificaciones que recibió la película)	Float64

- Pasos de Preprocesamiento

  Nota: El proceso descrito a continuación se puede ver a detalle en la siguiente liga de Google Colab: https://colab.research.google.com/drive/1bcVz_nOkUFUtlEYD-u2Ihtq7wgQMGZaa?usp=sharing

1. Selección de variables relevantes de cada archivo.

Archivo	**Columnas o Variables**
Keywords	Movie ID, Keywords
Credits	Movie ID, Cast, Crew
Movies	Movie ID, Genres, Original Title, Overview, Production Companies

2. Convertir la variable Movie ID del archivo Movies a Int64 para que esté en el mismo formato que las variables Movie ID de los archivos Keywords y Credits.

3. Creación de un data frame unificando los tres archivos (Credits, Keywords, Movies) con las variables o columnas seleccionadas. La unión se realiza en base a la columna Movie ID de cada archivo.

4. Búsqueda de registros vacíos en las variables.

Variable	**Registros Vacíos**
Production Companies	4
Overview	995

5. Llenado por conocimiento de los registros vacíos.

   Como la variable Production Companies presenta muy pocos registros vacíos, se investigaron los registros faltantes y se llenaron sus registros.

6. Eliminación de un registro vacío

   La variable Production Companies presenta un registro vacío, sin embargo, el nombre de la película está en caracteres chinos, por lo que no se entiende el nombre de la película y no se puede hacer la investigación del registro faltante. Como solo es un registro, no hay mayor complicación en eliminarlo.

7. Los valores faltantes de la variable Overview no tienen mayor afectación en el algoritmo de recomendación por lo que se agrega una lista vacía a cada registro, ya que estas películas no tienen público su overview en el idioma en el que se está trabajando (películas sólo publicadas en China).

8. Se convierten a lista los strings de las variables Genres, Keywords y Production Companies para un mejor rendimiento del algoritmo de recomendación.

9. De la variable Crew se obtiene el string del nombre del director y se convierte a lista para un mejor rendimiento del algoritmo de recomendación.

10. De la variable Cast se obtienen los nombres de los actores o personajes (película animada) y se convierten a lista para un mejor rendimiento del algoritmo de recomendación.

11. Las palabras de la variable Overview en formato string se convierten en lista para un mejor rendimiento del algoritmo de recomendación.

12. El conjunto de datos después del preprocesamiento queda de la siguiente manera.

	id	keywords	genres	original_title	overview	production_companies	cast	crew
0	862	[jealousy, toy, boy, friendship, friends, riva...	[Animation, Comedy, Family]	Toy Story	[Led, by, Woody,, Andy's, toys, live, happily,...	[Pixar Animation Studios]	[Tom Hanks, Tim Allen, Don Rickles, Jim Varney...	[John Lasseter]
1	8844	[board game, disappearance, based on children'...	[Adventure, Fantasy, Family]	Jumanji	[When, siblings, Judy, and, Peter, discover, a...	[TriStar Pictures, Teitler Film, Interscope Co...	[Robin Williams, Jonathan Hyde, Kirsten Dunst,...	[Joe Johnston]
2	15602	[fishing, best friend, duringcreditsstinger, o...	[Romance, Comedy]	Grumpier Old Men	[A, family, wedding, reignites, the, ancient, ...	[Warner Bros., Lancaster Gate]	[Walter Matthau, Jack Lemmon, Ann-Margret, Sop...	[Howard Deutch]
3	31357	[based on novel, interracial relationship, sin...	[Comedy, Drama, Romance]	Waiting to Exhale	[Cheated, on,, mistreated, and, stepped, on,, ...	[Twentieth Century Fox Film Corporation]	[Whitney Houston, Angela Bassett, Loretta Devi...	[Forest Whitaker]
4	11862	[baby, midlife crisis, confidence, aging, daug...	[Comedy]	Father of the Bride Part II	[Just, when, George, Banks, has, recovered, fr...	[Sandollar Productions, Touchstone Pictures]	[Steve Martin, Diane Keaton, Martin Short, Kim...	[Charles Shyer]

- Exploración Inicial

  Nota: El proceso descrito a continuación se puede ver a detalle en la siguiente liga de Google Colab: https://colab.research.google.com/drive/1bcVz_nOkUFUtlEYD-u2Ihtq7wgQMGZaa?usp=sharing

  Dimensiones:

  Número de Filas: 4662
  Número de Columnas: 8

  Variables:

  id
  keywords
  genres
  original_title
  overview
  production_companies
  cast
  crew

  Tipos de Datos:

  id (Int64)
  keywords (Object)

genres (Object)
original_title (Object)
overview (Object)
production_companies (Object)
cast (Object)
crew (Object)

Porcentaje Datos Faltantes:

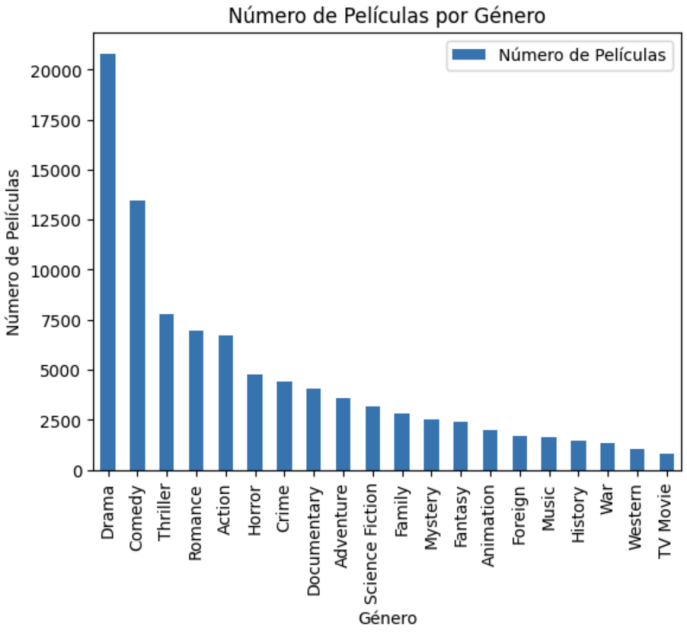
id (0%)
keywords (0%)
genres (0%)
original_title (0%)
overview (0%)
production_companies (0%)
cast (0%)
crew (0%)

- Análisis del Conjunto de Datos

  Nota: El proceso descrito a continuación se puede ver a detalle en la siguiente liga de Google Colab: https://colab.research.google.com/drive/1bcVz_nOkUFUtlEYD-u2Ihtq7wgQMGZaa?usp=sharing

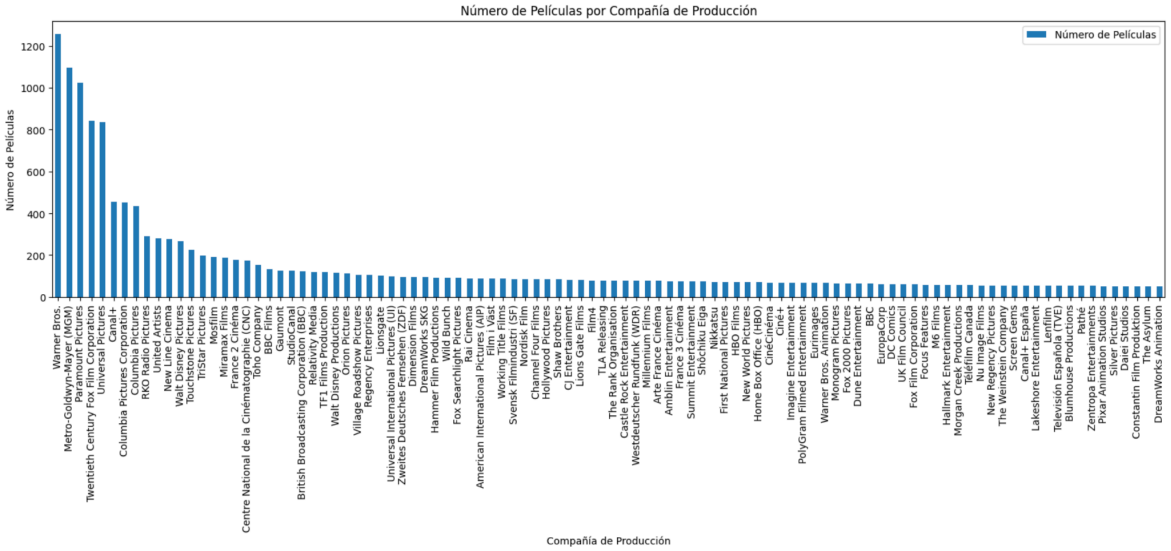
  - genres

    Son 20 géneros de películas distintos. Se muestra una gráfica del número de películas por género.



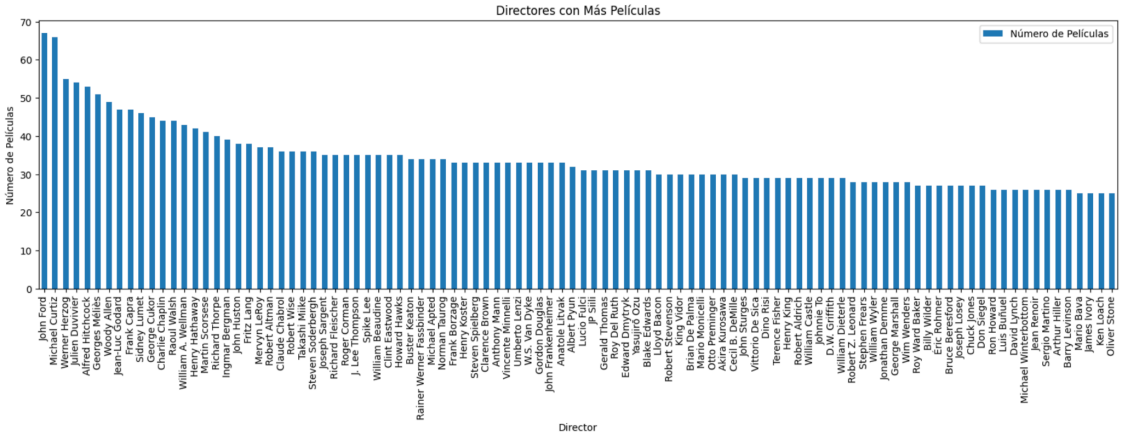
- production_companies

  Hay una variedad bastante grande de compañías de producción de películas. Se muestra una gráfica con las 100 compañías de producción con más películas en orden descendente.



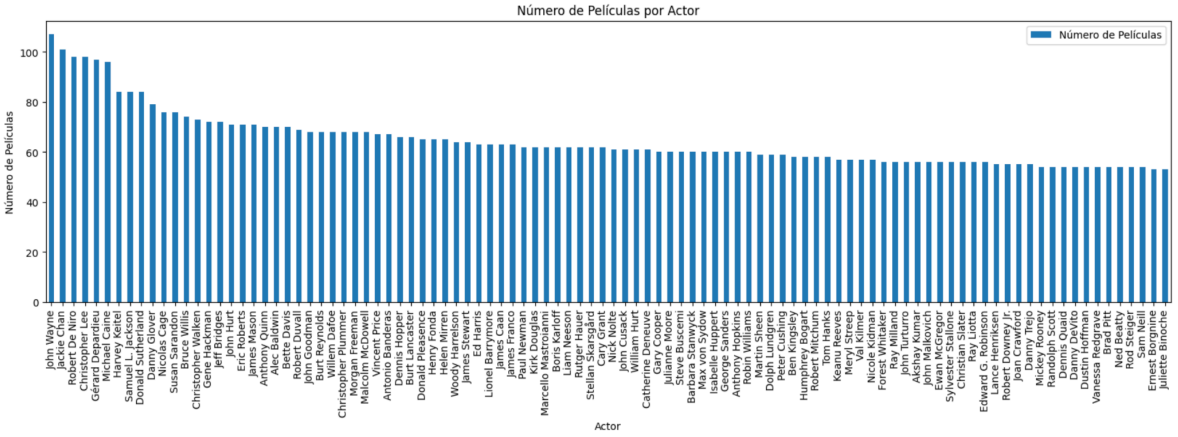
- crew

  Hay una variedad bastante grande de directores de películas. Se muestra una gráfica con los 100 directores con más películas en orden descendente.



- cast

  Hay una variedad bastante grande de actores y personajes (películas animadas). Se muestra una gráfica con los 100 actores o personajes (películas animadas) con más películas en orden descendente.



- id

  Todas las películas tienen un ID distinto, por lo que se tienen 4662 valores diferentes.

- original_title

  Todas las películas tienen títulos distintos, por lo que se tienen 4662 valores diferentes.

- keywords

  Son palabras clave de la trama de la película, por lo que se tiene una lista con un vocabulario distinto por película.

- overview

  Son palabras de la sinopsis de la película, por lo que se tiene una lista con un vocabulario distinto por película.

2. Algoritmo de Recomendación Básico

- Descripción del Algoritmo de Recomendación Básico

  Similitud de Coseno

  La recomendación de sistema por similitud de coseno es un enfoque utilizado en sistemas de recomendación para calcular la similitud entre dos elementos basándose en sus características. En el contexto de recomendación de películas, por ejemplo, se pueden utilizar las descripciones o metadatos de las películas (como género, actores, director, etc.) como características para calcular la similitud entre ellas.

  El cálculo de la similitud de coseno se basa en el concepto de espacio vectorial, donde cada elemento (en este caso, una película) se representa como un vector en un espacio multidimensional, donde cada dimensión corresponde a una característica de la película.

  El algoritmo de similitud de coseno mide el ángulo entre dos vectores, lo que refleja la similitud direccional entre ellos. Cuanto más cercano estén los vectores en dirección (es decir, cuanto menor sea el ángulo entre ellos), mayor será la similitud de coseno y, por lo tanto, mayor será la similitud entre los elementos. Para calcular la similitud de coseno entre dos elementos, se utiliza la fórmula:

  $$\text{similitud}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}$$

  Donde:

  - $\mathbf{A} \cdot \mathbf{B}$ es el producto punto entre los vectores $\mathbf{A}$ y $\mathbf{B}$.
  - $\|\mathbf{A}\|$ y $\|\mathbf{B}\|$ son las magnitudes (normas) de los vectores $\mathbf{A}$ y $\mathbf{B}$ respectivamente.

- Implementación

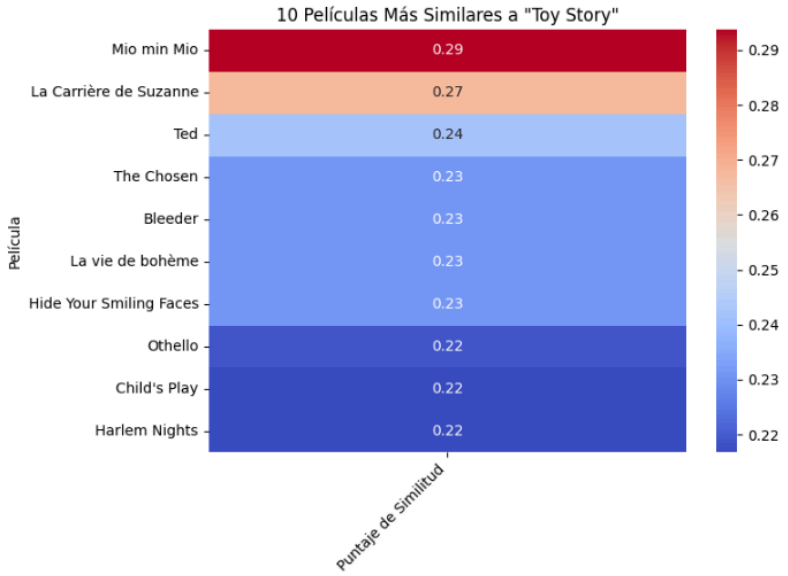
  En el contexto de un sistema de recomendación de películas, la similitud de coseno se calcula entre las representaciones vectoriales de las películas (por ejemplo, **utilizando las frecuencias de las palabras clave extraídas de las descripciones de las películas**) para determinar qué tan similares son las películas entre sí. Esto se utiliza para encontrar películas similares a una película de entrada específica y, así, hacer recomendaciones personalizadas.

  En términos generales, no hay un valor de similitud de coseno específico considerado "bueno" para aceptar una recomendación. La evaluación de la similitud de coseno depende del contexto del problema y las preferencias del usuario. Se espera que las películas recomendadas tengan una similitud alta entre sí y en comparación con otras películas. Es esencial evaluar tanto los valores numéricos como la relevancia cualitativa de las recomendaciones, considerando las características importantes para el usuario y el contexto del dominio.

- Evaluación, Resultados y Análisis

  Para nuestro ejemplo, utilizaremos la película “Toy Story” como referencia, el algoritmo nos deberá arrojar 10 recomendaciones de películas acorde al título y características de la misma.

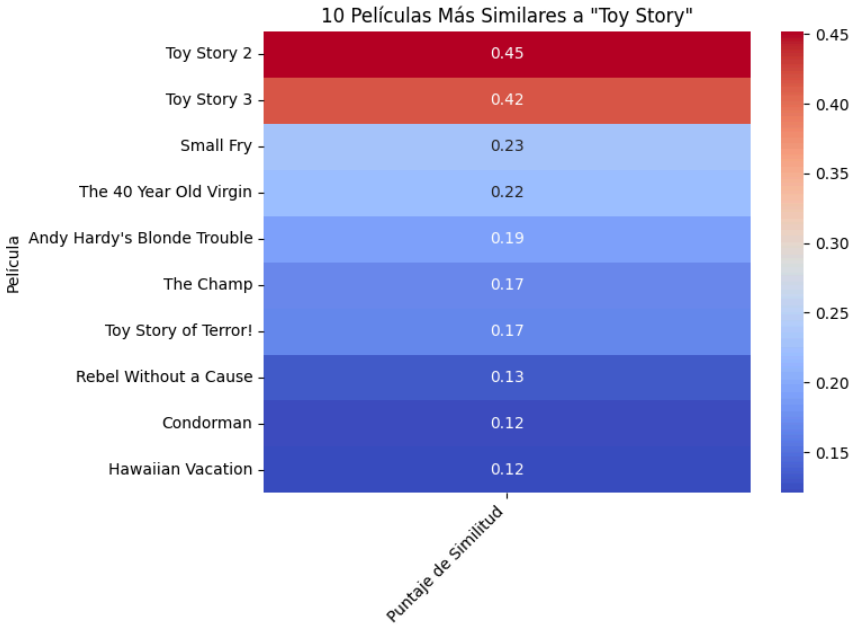
  Durante nuestra primera iteración, **al utilizar sólo las palabras clave del dataset**, el resultado no fue el esperado, pues las recomendaciones no nos hacían sentido (falsos positivos):



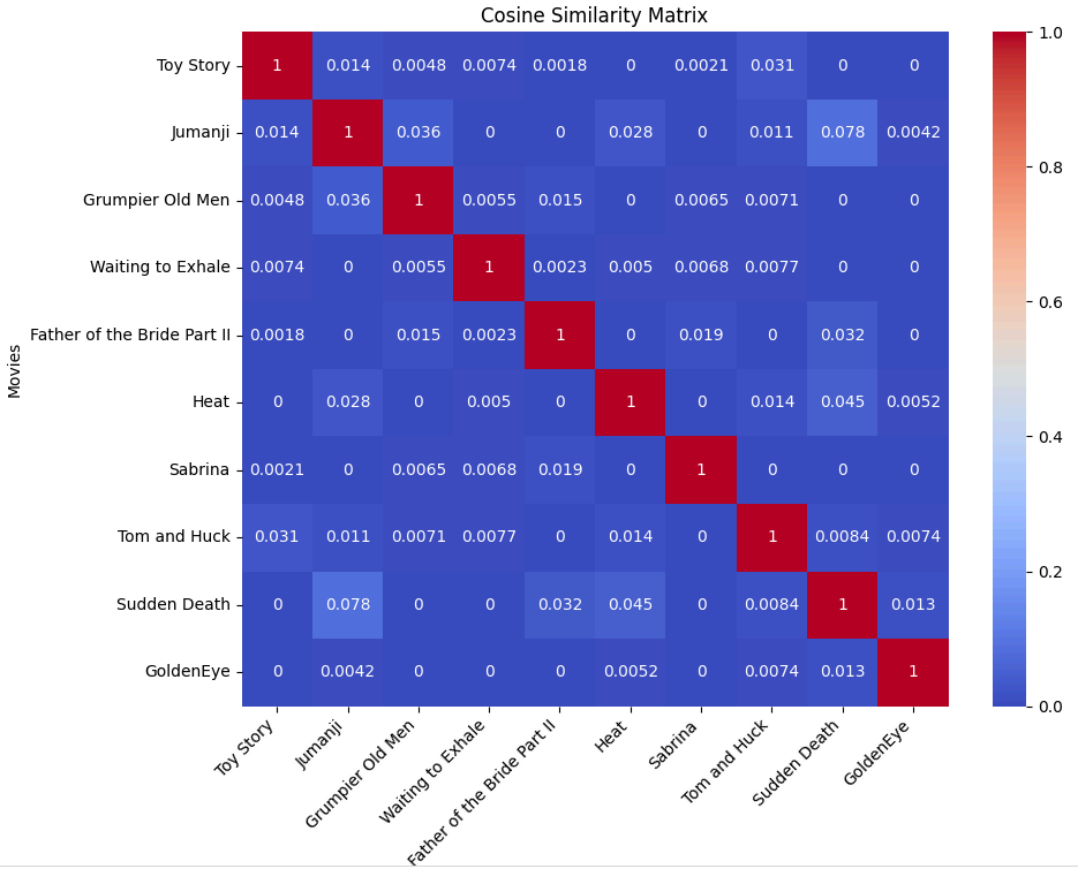
  Al analizar el dataset, se llegó a la conclusión de **enriquecer la columna de palabras clave, con otra metadata de la película como el género, la sinopsis, el reparto y el staff**.

```
complete_movies_df['collection'] = complete_movies_df['keywords'] +
complete_movies_df['genres'] + complete_movies_df['overview'] +
complete_movies_df['cast'] + complete_movies_df['crew']
```

  El resultado al hacer esta modificación, fue más preciso que durante la iteración anterior:

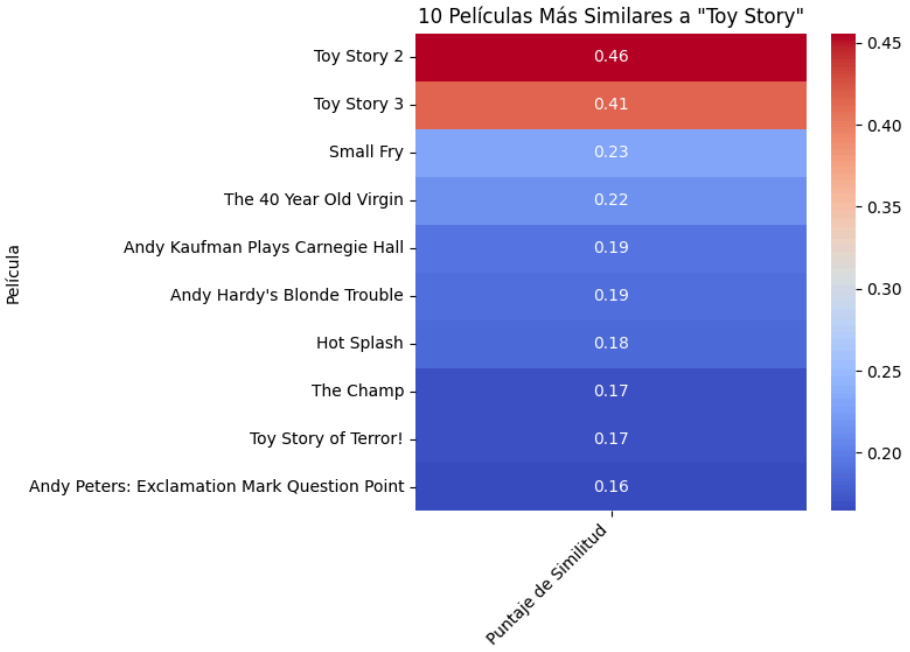


Como medida comparativa, graficamos también el heatmap para otras 10 películas, para observar qué tan despegados están los coeficientes en películas que no se recomiendan:



El algoritmo se tuvo que recortar a utilizar solamente 25,000 registros debido a la limitante de RAM al utilizar Google Colab.

Al utilizar el dataset completo (46627 filas × 3 columnas) en una PC local con mayores recursos, las recomendaciones cambiaron ligeramente.



3. Algoritmo de Recomendación Avanzado

Implementación de al menos un algoritmo de recomendación avanzados (por ejemplo, factorización matricial, enfoques basados en aprendizaje profundo)

- Descripción del Algoritmo de Recomendación Avanzado

  El algoritmo de recomendación avanzado que implementamos utiliza una red neuronal profunda, específicamente un autoencoder, para predecir las calificaciones de películas que los usuarios podrían dar, basándose en sus preferencias históricas.

  Estructura del Modelo

  - Capa de Entrada (Input Layer):
    - Descripción: Cada usuario se representa por un vector de calificaciones de películas.
    - Dimensión: El número de nodos en la capa de entrada es igual al número de películas en el dataset (denotado como n).
  - Capa Oculta (Hidden Layer):
    - Descripción: Una capa intermedia que intenta aprender una representación más compacta y significativa de las preferencias del usuario.
    - Función de Activación: Sigmoide
    - Número de Neuronas: En este caso, 256 nodos.
  - Capa de Salida (Output Layer):
    - Descripción: Esta capa trata de reconstruir las calificaciones originales del usuario.
    - Número de Neuronas: Igual al número de películas (n).
    - Función de Activación: Lineal, ya que estamos prediciendo calificaciones.
  - Función de Pérdida:
    - Utilizamos el Error Cuadrático Medio (MSE) para evaluar la discrepancia entre las calificaciones predichas y las reales:

- Implementación
  1. Carga y Preprocesamiento de Datos
     a. Se utilizó un dataset más grande de MovieLens que incluye calificaciones de películas dadas por diferentes usuarios.
     b. Los datos fueron transformados en una tabla pivote con usuarios en las filas, películas en las columnas y las calificaciones como valores.
  2. Normalización de Datos
     a. Se aplicó la normalización Min-Max para escalar las calificaciones en un rango de 0 a 1.
  3. División en Conjuntos de Entrenamiento y Prueba
     a. Los datos fueron divididos en conjuntos de entrenamiento y prueba con un 80% y 20% respectivamente.
  4. Construcción del Modelo de Red Neuronal
     a. Se definió un modelo secuencial de Keras con capas densas.

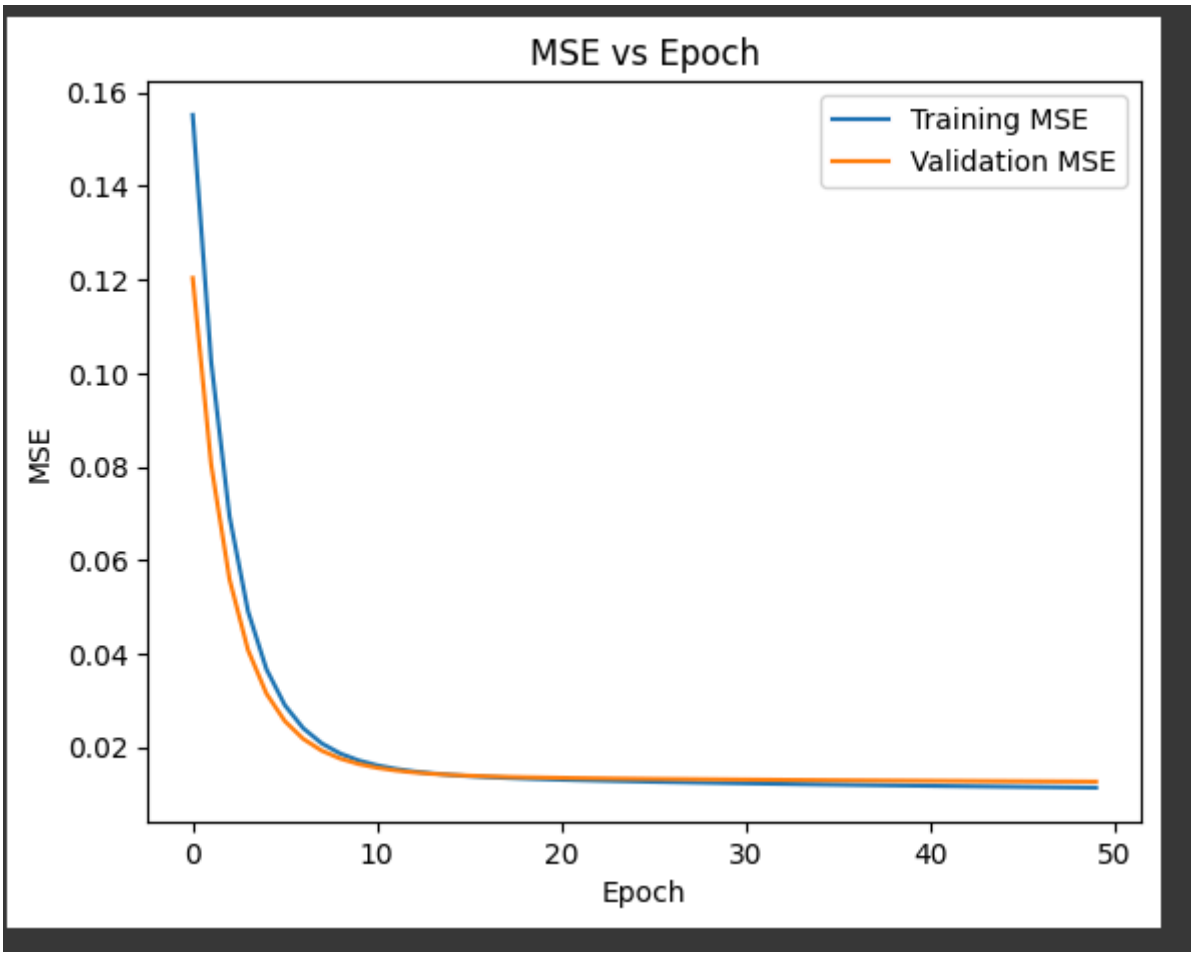
    b. La capa de entrada corresponde al número de características (películas).
    c. Se añadió una capa oculta con 256 nodos y una función de activación sigmoide.
    d. La capa de salida tiene el mismo número de nodos que el número de películas.
5. Compilación y Entrenamiento del Modelo
    a. Se utilizó el optimizador Adam y una tasa de aprendizaje de 0.0001.
    b. La función de pérdida es el error cuadrático medio (MSE).
    c. El modelo fue entrenado por 50 épocas con un tamaño de lote de 64.
6. Evaluación del Modelo
    a. Se graficó el error cuadrático medio (MSE) en el conjunto de entrenamiento y validación para cada época.
7. Generación de Recomendaciones
    a. Se define una función para obtener las recomendaciones de películas para un usuario específico basándose en las predicciones del modelo.

- <u>Evaluación</u>

  Identifica y justifica las métricas de evaluación utilizadas para evaluar el desempeño de los sistemas de recomendación

  Para evaluar el desempeño del sistema de recomendación, utilizamos las siguientes métricas:

  1. Error Cuadrático Medio (MSE)
      a. Esta métrica mide el promedio de los cuadrados de los errores o desviaciones, es decir, la diferencia entre los valores predichos por el modelo y los valores reales.
      b. Un MSE más bajo indica un mejor ajuste del modelo a los datos de entrenamiento.
  2. Curva de Aprendizaje
      a. Al graficar la pérdida del entrenamiento y la validación a lo largo de las épocas, podemos observar cómo mejora el modelo y si hay problemas de sobreajuste o sobreajuste.



- Resultados y Análisis

  Enlista la menos 3 recomendaciones donde se muestren los resultados obtenidos

  Al evaluar el modelo, obtenemos las siguientes recomendaciones para un usuario de prueba (user_id=10):

  1. Recomendación 1
     a. Título: "Secret of Roan Inish, The (1994)"
     b. Predicted Rating: 0.5054865479469299
  2. Recomendación 2
     a. Título: "Hate (Haine, La) (1995)"
     b. Predicted Rating: 0.44272491335868835
  3. Recomendación 3
     a. Título: "Miracle on 34th Street (1994)"
     b. Predicted Rating: 0.4089670777320862

  Estas recomendaciones se basan en las preferencias del usuario y en cómo el modelo ha aprendido a predecir sus calificaciones potenciales para diferentes películas.

  El código implementa un sistema de recomendación avanzado basado en una red neuronal profunda utilizando un auto encoder. Las recomendaciones generadas parecen razonables y alineadas con las preferencias esperadas de los usuarios basadas en sus calificaciones históricas.

En cuanto a la eficiencia del modelo, la red neuronal profunda (autoencoder) es efectiva en capturar patrones complejos en los datos de calificaciones, ofreciendo recomendaciones personalizadas, aunque tiene algunas limitaciones como:

- El modelo podría beneficiarse de más datos y recursos computacionales para mejorar la precisión y eficiencia.
- Se pueden explorar arquitecturas de red más complejas o técnicas de regularización para mejorar el rendimiento y evitar el sobreajuste.

Algunas mejoras que se podrían aplicar serían:

- Incluir más características contextuales (por ejemplo, metadatos de películas) para enriquecer el modelo.
- Experimentar con diferentes optimizadores y funciones de pérdida.
- Evaluar el modelo utilizando métricas adicionales como MAE (Error Absoluto Medio) e implementar validación cruzada para una evaluación más robusta.